EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	61	(virtual adj counter)	US-PGPUB; USPAT	OR	ON	2007/06/23 14:33
L2	37	(virtual adj time$2) with (lag$4 speed drift$4)	US-PGPUB; USPAT	OR	ON	2007/06/23 15:22
L3	40	reduc$ with migrat$4 with (drift$4 penal$4)	US-PGPUB; USPAT	OR	ON	2007/06/23 15:22
L4	10	minimiz$ with migrat$4 with (drift$4 penal$4)	US-PGPUB; USPAT	OR	ON	2007/06/23 15:23
L5	5754	minimiz$ with (drift$4 penal$4)	US-PGPUB; USPAT	OR	ON	2007/06/23 15:23
L6	2	(reduc$4 minimiz$) with (drift$4 penal$4) with ((virtual adj (system machine)) VM VMM)	US-PGPUB; USPAT	OR	ON	2007/06/23 15:24
L7	181	((Virtual adj machine) VM VMM) and 703/23-28.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/23 15:25
L8	159	((Virtual adj machine) VM VMM) and 703/13-22.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/23 15:25
L9	312	L7 L8	US-PGPUB; USPAT	OR	ON	2007/06/23 15:25
L10	8	("5103394" | "5666519" | "5678028" | "5737579" | "5761477" | "5815688" | "6047381").PN. OR ("6882968").URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2007/06/23 15:48
L11	4	scala$5 with (virtual adj (time$2))	US-PGPUB; USPAT; USOCR	OR	OFF	2007/06/23 15:58
L12	38	scala$5 with (virtual adj (machine))	US-PGPUB; USPAT; USOCR	OR	ON	2007/06/23 15:58
L18	15	((apparent simulat$4 virtual) adj (time clock)) with ((real wall hardware) adj (time clock)) with (lag$4 drift$4 behind "catch-up" delay$4)	US-PGPUB; USPAT	OR	ON	2007/06/23 16:12
L20	182	(suspend$4 halt$4) with (virtual adj machine)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 16:23

EAST Search History

L21	19	(US-20070033589-$ or US-20060075402-$ or US-20050204357-$ or US-20060130059-$ or US-20050027500-$).did. or (US-6550015-$ or US-6349388-$ or US-4814975-$ or US-5774479-$ or US-5488713-$ or US-5621912-$ or US-5095427-$ or US-4812967-$ or US-6882968-$ or US-7136800-$ or US-7146305-$).did. or (US-6550015-$ or US-20020056076-$ or EP-419723-$).did.	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 16:54
L22	6	L21 and timers	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 16:59
L23	735	virtual adj time	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 17:01
L24	397	L23 and ((real hardware) adj time)	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 17:00
L25	397	L23 and ((wall real hardware) adj time)	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 17:00
L26	123	virtual adj time with (((wall real hardware) adj time))	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 17:02
L27	3	L26 and ((virtual adj (machine system)) VM VMM hypervisor)	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 17:26

L28	47	US-5437033-$.DID. OR US-4811276-$.DID. OR US-5295265-$.DID. OR US-5023771-$.DID. OR US-5355470-$.DID. OR US-6412035-$.DID. OR US-4814975-$.DID. OR US-6373846-$.DID. OR US-5898855-$.DID. OR US-6208661-$.DID. OR US-6961806-$.DID. OR US-6996748-$.DID. OR US-7069413-$.DID. OR US-7082598-$.DID. OR US-7089377-$.DID. OR US-7111086-$.DID. OR US-7111145-$.DID. OR US-7124327-$.DID. OR US-7127548-$.DID. OR US-7149843-$.DID. OR US-7155558-$.DID. OR US-7191440-$.DID. OR US-20020172202-$.DID. OR US-20060005200-$.DID. OR US-20030217250-$.DID. OR US-20040003323-$.DID. OR US-20040003324-$.DID. OR US-20040117539-$.DID. OR US-20040123288-$.DID. OR US-20040205203-$.DID. OR US-20040268347-$.DID. OR US-20050060702-$.DID. OR US-20050060703-$.DID. OR US-20050071840-$.DID. OR US-20050080753-$.DID. OR US-20050080934-$.DID. OR US-20050080937-$.DID. OR US-20050081199-$.DID. OR US-20050132362-$.DID. OR US-20050132365-$.DID. OR US-20050216920-$.DID. OR US-20050223377-$.DID. OR US-20050289542-$.DID. OR US-20060004554-$.DID. OR US-20060004667-$.DID. OR US-20060005003-$.DID. OR US-20060005184-$.DID.	US-PGPUB; USPAT	OR	OFF	2007/06/23 17:28
S1	56	virtual adj timer	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 16:19

EAST Search History

S2	2	"09/247,876"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/19 14:56
S3	322	703/19.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/19 15:06
S4	596	718/1.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/19 15:08
S7	18	("3881156" | "4287562" | "4879733" | "4912734" | "4926319" | "4942522" | "4952367" | "5042005" | "5089955" | "5220661" | "5233573" | "5325341" | "5363499" | "5491815" | "5664167" | "5724399" | "5740451" | "5975739").PN. OR ("6550015"). URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2007/06/20 18:23
S8	510	(((speed catch) adj up) accelerat$4) with ((virtual adj machine) VMM VM)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 16:56
S9	63	S8 and timer	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 14:14

EAST Search History

S10	40	(US-20020129338-$ or US-20020056076-$).did. or (US-6108309-$ or US-6522985-$ or US-5247653-$ or US-5550760-$ or US-6117181-$ or US-6725188-$ or US-6173249-$ or US-6134516-$ or US-5287461-$ or US-6792460-$ or US-6934755-$ or US-6240529-$ or US-5784552-$ or US-5937179-$ or US-6618839-$ or US-6901581-$ or US-5621912-$ or US-6795966-$). did.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 14:54
S11	510	(((speed catch) adj up) accelerat$4) with ((virtual adj machine) VMM VM)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 17:09
S12	73	S11 and interrupt	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 15:07
S13	9	S10 and (VM VMM (virtual adj machine))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 14:54
S14	2	(catch adj up) with virtual with (timer timming)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 15:08
S15	2	(catch adj up) same (virtual with (timer timming))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 15:08

EAST Search History

S16	18	(catch adj up) and (virtual with (timer timming))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 15:10
S17	598	718/1.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 15:11
S18	71	S17 and timer	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/21 15:12
S20	23	US-20010021969-$.DID. OR US-20010027511-$.DID. OR US-20010027527-$.DID. OR US-20010037450-$.DID. OR US-20020007456-$.DID. OR US-20020023032-$.DID. OR US-20020147916-$.DID. OR US-20020166061-$.DID. OR US-20020169717-$.DID. OR US-20030018892-$.DID. OR US-20030074548-$.DID. OR US-20030115453-$.DID. OR US-20030126442-$.DID. OR US-20030126453-$.DID. OR US-20030159056-$.DID. OR US-20030188179-$.DID. OR US-20030196085-$.DID. OR US-20040117539-$.DID. OR US-3699532-$.DID. OR US-3996449-$.DID. OR US-4037214-$.DID. OR US-4162536-$.DID. OR US-4207609-$.DID.	US-PGPUB; USPAT	OR	OFF	2007/06/21 17:00

S21	41	US-4276594-$.DID. OR US-4278837-$.DID. OR US-4307447-$.DID. OR US-4319233-$.DID. OR US-4319323-$.DID. OR US-4347565-$.DID. OR US-4366537-$.DID. OR US-4403283-$.DID. OR US-4419724-$.DID. OR US-4430709-$.DID. OR US-4521852-$.DID. OR US-4759064-$.DID. OR US-4795893-$.DID. OR US-4802084-$.DID. OR US-4825052-$.DID. OR US-4907270-$.DID. OR US-4907272-$.DID. OR US-4910774-$.DID. OR US-4975836-$.DID. OR US-5007082-$.DID. OR US-5022077-$.DID. OR US-5075842-$.DID. OR US-5079737-$.DID. OR US-5187802-$.DID. OR US-5230069-$.DID. OR US-5237616-$.DID. OR US-5255379-$.DID. OR US-5287363-$.DID. OR US-5293424-$.DID. OR US-5295251-$.DID. OR US-5317705-$.DID. OR US-5319760-$.DID. OR US-5361375-$.DID. OR US-5386552-$.DID. OR US-5421006-$.DID. OR US-5434999-$.DID. OR US-5437033-$.DID. OR US-5442645-$.DID. OR US-5455909-$.DID. OR US-5459867-$.DID. OR US-5459869-$.DID.	US-PGPUB; USPAT	OR	OFF	2007/06/21 17:01

S22	84	US-5473692-$.DID. OR US-5479509-$.DID. OR US-5504922-$.DID. OR US-5506975-$.DID. OR US-5511217-$.DID. OR US-5522075-$.DID. OR US-5528231-$.DID. OR US-5533126-$.DID. OR US-5555385-$.DID. OR US-5555414-$.DID. OR US-5560013-$.DID. OR US-5564040-$.DID. OR US-5566323-$.DID. OR US-5568552-$.DID. OR US-5574936-$.DID. OR US-5582717-$.DID. OR US-5604805-$.DID. OR US-5606617-$.DID. OR US-5615263-$.DID. OR US-5628022-$.DID. OR US-5633929-$.DID. OR US-5657445-$.DID. OR US-5668971-$.DID. OR US-5684948-$.DID. OR US-5706469-$.DID. OR US-5717903-$.DID. OR US-5720609-$.DID. OR US-5721222-$.DID. OR US-5729760-$.DID. OR US-5737604-$.DID. OR US-5737760-$.DID. OR US-5740178-$.DID. OR US-5752046-$.DID. OR US-5757919-$.DID. OR US-5764969-$.DID. OR US-5796835-$.DID. OR US-5796845-$.DID. OR US-5805712-$.DID. OR US-5809546-$.DID. OR US-5825875-$.DID. OR US-5825880-$.DID. OR US-5835594-$.DID. OR US-5852717-$.DID. OR US-5854913-$.DID. OR US-5867577-$.DID. OR US-5872994-$.DID. OR US-5890189-$.DID. OR US-5900606-$.DID. OR US-5901225-$.DID. OR US-5903752-$.DID. OR US-5919257-$.DID. OR US-5935242-$.DID. OR US-5935247-$.DID. OR US-5937063-$.DID. OR US-5944821-$.DID. OR US-5953502-$.DID. OR US-5956408-$.DID. OR US-5970147-$.DID. OR US-5978475-$.DID. OR	US-PGPUB; USPAT	OR	OFF	2007/06/21 17:03

S24	51	US-6192455-$.DID. OR US-6199152-$.DID. OR US-6205550-$.DID. OR US-6212635-$.DID. OR US-6222923-$.DID. OR US-6249872-$.DID. OR US-6252650-$.DID. OR US-6269392-$.DID. OR US-6272533-$.DID. OR US-6272637-$.DID. OR US-6275933-$.DID. OR US-6282650-$.DID. OR US-6282651-$.DID. OR US-6282657-$.DID. OR US-6292874-$.DID. OR US-6301646-$.DID. OR US-6308270-$.DID. OR US-6314409-$.DID. OR US-6321314-$.DID. OR US-6327652-$.DID. OR US-6330670-$.DID. OR US-6339815-$.DID. OR US-6339816-$.DID. OR US-6357004-$.DID. OR US-6363485-$.DID. OR US-6374286-$.DID. OR US-6374317-$.DID. OR US-6378068-$.DID. OR US-6378072-$.DID. OR US-6389537-$.DID. OR US-6397242-$.DID. OR US-6397379-$.DID. OR US-6412035-$.DID. OR US-6421702-$.DID. OR US-6435416-$.DID. OR US-6445797-$.DID. OR US-6463535-$.DID. OR US-6463537-$.DID. OR US-6499123-$.DID. OR US-6505279-$.DID. OR US-6507904-$.DID. OR US-6529909-$.DID. OR US-6535988-$.DID. or "6557104". pn. or US-6560627-$.DID. OR US-6609199-$.DID. OR US-6615278-$.DID. OR US-6633963-$.DID. OR US-6633981-$.DID. OR US-6651171-$.DID. OR US-6678825-$.DID.	US-PGPUB; USPAT	OR	OFF	2007/06/21 17:04
S25	63	(S20 or S21 or S22 or S24) and (timer VMM VM)	US-PGPUB; USPAT	OR	OFF	2007/06/21 17:08
S26	1	virtual with timer with latenc$4	US-PGPUB; USPAT	OR	OFF	2007/06/21 17:10

EAST Search History

S27	4	(increas$4 decreas$4) with ((virtual adj machine) VMM VM) with latenc$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 17:11
S28	211	((virtual adj machine) VMM VM) with timer	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/22 13:56
S29	477	((virtual adj (system machine)) VMM VM) with ((catch adj up) accelerat$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/22 14:21
S30	852	((virtual adj (system machine)) VMM VM) with (frequenc$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/22 14:53
S31	62	S30 and timer	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/22 14:24
S32	254	S30 and (index$2 counter timer)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/22 14:24
S33	70	((virtual adj (system machine)) VMM VM) with (interrupt event) with (queue)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/22 15:16

EAST Search History

S34	6953	"718".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/22 15:11
S35	1198	S34 and timer	US-PGPUB; USPAT	OR	ON	2007/06/22 15:12
S36	514	718/106.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/22 15:12
S37	87	S36 and timer	US-PGPUB; USPAT	OR	ON	2007/06/22 15:12
S38	1481	((virtual adj (system machine)) VMM VM) with (restart resume start)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/22 15:18
S39	196	S38 and S34	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/22 15:22
S40	579	"703".clas. and (virtual adj2 (computer system machine))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/22 22:13
S41	37	((virtual adj machine) VMM VM) with drift	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 10:54
S42	11	(timothy with Mann).in.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 12:16

EAST Search History

S43	44	vmware.as.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 12:21
S44	4	vmware.as. and timer	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 12:23
S45	13	(virtual adj (system machine))with (event adj queu$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 12:41
S49	181	((Virtual adj machine) VM VMM) and 703/23-28.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/23 12:42
S50	159	((Virtual adj machine) VM VMM) and 703/13-22.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/23 15:25
S51	20	(Virtual adj (system event timer)) and 703/23-28.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/23 12:43
S52	30	(Virtual adj (system event timer)) and 703/13-22.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/23 12:43

VMTN VMware Technology Network

vmware

Technical Resources

To view a paper directly, click the icon.

Filter by Product:	Filter by Area:	Filter by Publisher:
All Products	All Areas	All Publishers
VMware Infrastructure 3	Business Continuity (Backup & HA)	VMware
ESX Server 2.x	Desktop Manageability & Security	Network Appliance
VirtualCenter 1.x	Development & Test	RapidApp

For multiple selection, CTRL + Click

Filter Results

Title	Publisher	Rating	Revised
Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor	VMware		06/24/2001
Memory Resource Management in VMware ESX Server	VMware		12/08/2002
NIC Teaming IEEE 802.3ad	VMware		08/11/2003
An Analysis of Disk Performance in VMware ESX Server Virtual Machines	VMware		10/26/2003
Configuring and Installing IBM BladeCenter	VMware		03/31/2004
Hyper-Threading Support in ESX Server 2.1	VMware		04/13/2004
Configuring and Installing HP Blade Servers	VMware		04/28/2004
Login Using Windows NT	VMware		05/13/2004
Streamlining Patch Testing and Deployment	VMware		06/15/2004
Virtual to Physical Documentation and Sample Configurations	VMware		06/18/2004
Converting Image Files into Virtual Machine Disks	VMware		07/13/2004

Building Virtual Infrastructure with VMware VirtualCenter	VMware	07/29/2004
NIC Bonding and VLANs on IBM BladeCenter	VMware	08/10/2004
Security	VMware	09/24/2004
Best Practices for ESX Server 2	VMware	11/04/2004
Systems Management	VMware	12/06/2004
Reduce Oracle Real Application Clusters Deployment Costs and Cycle Times	VMware	12/06/2004
HCL: VMware ESX Server Supported Backup Tools	VMware	12/09/2004
VLAN Solutions	VMware	12/10/2004
Representing Physical Machines in the Virtual World	VMware	12/23/2004
Isolating Performance Problems	VMware	12/23/2004
Using esxtop to Troubleshoot Performance Problems	VMware	12/23/2004
ESX Performance Tips and Tricks	VMware	02/07/2005
Managing Remote Access	VMware	02/13/2005
Enforcing Patch Management	VMware	02/13/2005
Best Practices for Setting up VMware ACE	VMware	02/13/2005

Reference & Planning for Virtualizing Citrix	VMware	02/22/2005
Login Using Active Directory	VMware	03/02/2005
Reference and Capacity Planning with Citrix Presentation Server (for ESX Server 2)	VMware	03/09/2005
Comparing the MUI, VirtualCenter, and vmkusage	VMware	03/18/2005

1 2 3 4 5 next page last page

 Legal Privacy Accessibility Site Index

RSS

VMTN VMware Technology Network

vmware

Technical Resources

To view a paper directly, click the icon.

Filter by Product:	Filter by Area:	Filter by Publisher:
All Products	All Areas	All Publishers
VMware Infrastructure 3	Business Continuity (Backup & HA)	VMware
ESX Server 2.x	Desktop Manageability & Security	Network Appliance
VirtualCenter 1.x	Development & Test	RapidApp

For multiple selection, CTRL + Click

Filter Results

Title	Publisher	Rating	Revised
Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor	VMware		06/24/2001
Memory Resource Management in VMware ESX Server	VMware		12/08/2002
NIC Teaming IEEE 802.3ad	VMware		08/11/2003
An Analysis of Disk Performance in VMware ESX Server Virtual Machines	VMware		10/26/2003
Configuring and Installing IBM BladeCenter	VMware		03/31/2004
Hyper-Threading Support in ESX Server 2.1	VMware		04/13/2004
Configuring and Installing HP Blade Servers	VMware		04/28/2004
Login Using Windows NT	VMware		05/13/2004
Streamlining Patch Testing and Deployment	VMware		06/15/2004

Virtual to Physical Documentation and Sample Configurations VMware 06/18/2004

Converting Image Files into Virtual Machine Disks VMware 07/13/2004

Building Virtual Infrastructure with VMware VirtualCenter VMware 07/29/2004

NIC Bonding and VLANs on IBM BladeCenter VMware 08/10/2004

Security VMware 09/24/2004

Best Practices for ESX Server 2 VMware 11/04/2004

Systems Management VMware 12/06/2004

Reduce Oracle Real Application Clusters Deployment Costs and Cycle Times VMware 12/06/2004

HCL: VMware ESX Server Supported Backup Tools VMware 12/09/2004

VLAN Solutions VMware 12/10/2004

Representing Physical Machines in the Virtual World VMware 12/23/2004

Isolating Performance Problems VMware 12/23/2004

Using esxtop to Troubleshoot Performance Problems VMware 12/23/2004

ESX Performance Tips and Tricks VMware 02/07/2005

Managing Remote Access VMware 02/13/2005

Enforcing Patch Management	VMware	02/13/2005
Best Practices for Setting up VMware ACE	VMware	02/13/2005
Reference & Planning for Virtualizing Citrix	VMware	02/22/2005
Login Using Active Directory	VMware	03/02/2005
Reference and Capacity Planning with Citrix Presentation Server (for ESX Server 2)	VMware	03/09/2005
Comparing the MUI, VirtualCenter, and vmkusage	VMware	03/18/2005

1 2 3 4 5 next page last page

 Legal Privacy Accessibility Site Index

RSS

Sat, 23 Jun 2007, 12:07:06 PM EST

Search Query Display

Edit an existing query or compose a new query in the Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Recent Search Queries		Results
#1	(virtual machine<in>metadata)	2221
#2	(((virtual machine<in>metadata))<AND>(catch-up<in>metadata))	0
#3	(virtual machine<in>metadata)	2221
#4	(virtual machine<in>metadata)	2221
#5	(accelerat*<or>speed<IN>metadata)	173380
#6	(virtual machine<in>metadata)	2221
#7	(((virtual machine<in>metadata))<AND>(accelerat*<or>speed<in>metadata))	278
#8	(((virtual machine<in>metadata))<AND>(accelerat*<or>speed<in>metadata))	278
#9	(virtual machine<in>metadata)	2221
#10	(((virtual machine<in>metadata))<AND>(restart*<or>resum*<or>start*<in>metadata))	218
#11	(virtual machine<in>metadata)	2221
#12	(((virtual machine<in>metadata))<AND>(suspend*<in>metadata))	5
#13	(((virtual machine<in>metadata))<AND>(suspend*<in>metadata))	5
#14	((event queu*<in>metadata) <and> (virtual machine<in>metadata))	1
#15	((time<in>metadata) <and> (accelerat*<in>metadata))<and>(virtual machine<in>metadata)	21
#16	((virtual machine<in>metadata) <and> (smp<in>metadata))	9
#17	(virtual timer<in>metadata)	0
#18	(mann t.<in>au)	22
#19	((mann<in>au) <and>(virtual machine<in>metadata))	0
	((mann<in>au) <and>(virtual machine<in>metadata))	0

#20		
#21	((mann<in>au))	507
#22	((((mann<in>au)))<AND>(virtual machine<in>metadata))	0
#23	((mann<in>au))	507
#24	((mann t.<in>au))	22
#25	((dependent<in>metadata) <and> (virtual machine<in>metadata))	29
#26	((virtual machine<in>metadata) <and> (queu* interrupt*<in>metadata))	0
#27	((queu* interrupt*<in>metadata))	0
#28	((queu* interrupt<in>metadata))	0
#29	((interrupt queue*<in>metadata))	0
#30	((interrupt queue<in>metadata))	0
#31	((interrupt <in>metadata))	2076
#32	((((interrupt <in>metadata)))<AND>(virtual machine<in>metadata))	13

Clear Session History

Help Contact Us Privacy & Security IEEE.org

Indexed by Inspec

Edit an existing query or compose a new query in the Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Sat, 23 Jun 2007, 1:41:49 PM EST

Search Query Display

Recent Search Queries		**Results**
#1	((virtual machine<in>metadata) <and> (drift<in>metadata))	2
#2	((virtual machine<in>metadata) <and> (suspend<in>metadata))<or> (migrat*<in>metadata)	4890
#3	((virtual machine<in>metadata) <and> (suspend<in>metadata))<and> (resume<in>metadata)	3
#4	((virtual machine<in>metadata) <and> (suspend*<in>metadata))<and> (resum*<in>metadata)	3
#5	((((virtual machine<in>metadata) <and> (suspend<in>metadata))<or> (migrat*<in>metadata))<AND> (timer<in>metadata))	1

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login
Search: The ACM Digital Library The Guide
+"virtual machine" +timer suspend resume queue

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used: virtual machine timer suspend resume queue

Found 457 of 204,472

Sort results by relevance
Display results expanded form

Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200
Best 200 shown

Result page: 1 2 3 4 5 6 7 8 9 10 next

Relevance scale

1 Operating system principles
Per Brinch Hansen
January 1973 Book
Publisher: Prentice-Hall, Inc.
Full text available: pdf(16.81 MB) Additional Information: full citation, abstract, references, citings, index terms

From the Preface

MAIN GOAL

This book tries to give students of computer science and professional programmers a general understanding of *operating systems*--the programs that enable people to share computers efficiently.

To make the sharing of a computer tolerable, an operating system must enforce certain rules of behavior on all its users. One would therefore expect the designers of operating systems to do their utmost to make them as s ...

2 Real-time convergence of Ada and Java™
Ben Brosgol, Brian Dobbing
September 2001 **ACM SIGAda Ada Letters , Proceedings of the 2001 annual ACM SIGAda international conference on Ada SIGAda '01**, Volume XXI Issue 4
Publisher: ACM Press
Full text available: pdf(181.39 KB) Additional Information: full citation, abstract, references, citings, index terms

Two independent recent efforts have defined extensions to the Java platform that intend to satisfy real-time requirements. This paper summarizes the major features of these efforts, compares them to each other and to Ada 95's Real-Time Annex, and argues that their convergence with Ada95 may serve to complement rather than compete with Ada in the real-time domain.

Keywords: Ada, Java, Real-Time, asynchrony, garbage collection, scheduling, threads

3 Mobile and distributed systems: Enabling Java mobile computing on the IBM Jikes research virtual machine
Giacomo Cabri, Letizia Leonardi, Raffaele Quitadamo
August 2006 **Proceedings of the 4th international symposium on Principles and practice of programming in Java PPPJ '06**
Publisher: ACM Press
Full text available: pdf(509.98 KB) Additional Information: full citation, abstract, references, index terms

Today's complex applications must face the distribution of data and code among different network nodes. Java is a wide-spread language that allows developers to build complex software, even distributed, but it cannot handle the migration of computations (i.e. threads), due to intrinsic limitations of many traditional JVMs. After analyzing the approaches in literature, this paper presents our research work on the IBM Jikes Research Virtual Machine: exploiting some of its innovative VM techniques, ...

Keywords: Java virtual machine, code mobility, distributed applications, thread persistence

4 VSched: Mixing Batch And Interactive Virtual Machines Using Periodic Real-time Scheduling
Bin Lin, Peter A. Dinda
November 2005 **Proceedings of the 2005 ACM/IEEE conference on Supercomputing SC '05**
Publisher: IEEE Computer Society
Full text available: pdf(5.91 MB) Additional Information: full citation, abstract, citings, index terms

We are developing Virtuoso, u system ,for distributed computing using virtual machines (VMs). Virtuoso must be uble to mix batch und interactive VMs on the same physical hardwure, while satisfiing constraint on re- sponsiveness und compute rates for each workload. VSched is the component of Virtuoso that provides this capability. VSched is an entirely user-level tool that interacts with the stock Linux kernel running below any type-11 virtual machine monitor to schedule VMs (indeed, any process) ...

5 A foundation for an efficient multi-threaded scheme system
Suresh Jagannathan, Jim Philbin
January 1992 **ACM SIGPLAN Lisp Pointers , Proceedings of the 1992 ACM conference on LISP and functional programming LFP '92**, Volume V Issue 1
Publisher: ACM Press
Full text available: pdf(1.19 MB) Additional Information: full citation, abstract, references, citings, index terms

We have built a parallel dialect of Scheme called STING that differs from its contemporaries in a number of important respects. STING is intended to be used as an operating system substrate for modern parallel programming languages. The basic concurrency management objects is STING are first-class lightweight threads of control and virtual processors (VPs). Unlike high-level concurrency structures, STING threads and VPs are not encumbered by complex synchronization protocols. ...

6 The Atomos transactional programming language
Brian D. Carlstrom, Austen McDonald, Hassan Chafi, JaeWoong Chung, Chi Cao Minh, Christos Kozyrakis, Kunle Olukotun
June 2006 **ACM SIGPLAN Notices , Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation PLDI '06**, Volume 41 Issue 6
Publisher: ACM Press
Full text available: pdf(244.69 KB) Additional Information: full citation, abstract, references, citings, index terms

Atomos is the first programming language with implicit transactions, strong atomicity, and a scalable multiprocessor implementation. Atomos is derived from Java, but replaces its synchronization and conditional waiting constructs with simpler transactional alternatives.The Atomos watch statement allows programmers to specify fine-grained watch sets used with the Atomos retry conditional waiting statement for efficient transactional conflict-driven wakeup even in transactional memory systems with ...

Keywords: conditional synchronization, java, multiprocessor architecture, transactional memory

7 Runtime dynamics in collaborative systems

Du Li, Richard R. Muntz
November 1999 **Proceedings of the International ACM SIGGROUP conference on Supporting group work GROUP '99**
Publisher: ACM Press
Full text available: pdf(1.53 MB) Additional Information: full citation, abstract, references, citings, index terms

The importance of supporting flexible roles and dynamic policies has long been recognized in the CSCW (Computer-Supported Cooperative Work) literature but unfortunately never with a systematic solution. This paper proposes a taxonomy of runtime dynamics in collaborative systems in general and discusses our solution in the COCA framework. Firstly, individual participants can take roles, drop roles, and switch between roles as a collaboration runs, Secondly, a role can be transferred from one ...

Keywords: CSCW, collaboration, collaborative systems, distributed computing, dynamic reconfiguration, runtime dynamics

8 Process migration
Dejan S. Milojičić, Fred Douglis, Yves Paindaveine, Richard Wheeler, Songnian Zhou
September 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 3
Publisher: ACM Press
Full text available: pdf(1.24 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Process migration is the act of transferring a process between two machines. It enables dynamic load distribution, fault resilience, eased system administration, and data access locality. Despite these goals and ongoing research efforts, migration has not achieved widespread use. With the increasing deployment of distributed systems in general, and distributed operating systems in particular, process migration is again receiving more attention in both research and product development. As hi ...

Keywords: distributed operating systems, distributed systems, load distribution, process migration

9 Multiple Instruction Stream Processor
Richard A. Hankins, Gautham N. Chinya, Jamison D. Collins, Perry H. Wang, Ryan Rakvic, Hong Wang, John P. Shen
May 2006 **ACM SIGARCH Computer Architecture News , Proceedings of the 33rd annual international symposium on Computer Architecture ISCA '06**, Volume 34 Issue 2
Publisher: IEEE Computer Society, ACM Press
Full text available: pdf(976.83 KB) Additional Information: full citation, abstract, index terms

Microprocessor design is undergoing a major paradigm shift towards multi-core designs, in anticipation that future performance gains will come from exploiting threadlevel parallelism in the software. To support this trend, we present a novel processor architecture called the Multiple Instruction Stream Processing (MISP) architecture. MISP introduces the sequencer as a new category of architectural resource, and defines a canonical set of instructions to support user-level inter-sequencer signali ...

10 Experience and evolution of concurrent Smalltalk
Yasuhiko Yokote, Mario Tokoro
December 1987 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications OOPSLA '87**, Volume 22 Issue 12
Publisher: ACM Press
Full text available: pdf(974.72 KB) Additional Information: full citation, abstract, references, citings, index terms

ConcurrentSmalltalk is an object-oriented concurrent programming language/system which has been running since late 1985. ConcurrentSmalltalk has the following features: Upper-compatibility with Smalltalk-80. Asynchronous method calls and CBox objects yield

concurrency. Atomic objects have the property of running one at a time so that it can serialize the many requests sent to it.

11 Xen and the art of virtualization
Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, Andrew Warfield
October 2003 **ACM SIGOPS Operating Systems Review , Proceedings of the nineteenth ACM symposium on Operating systems principles SOSP '03**, Volume 37 Issue 5
Publisher: ACM Press
Full text available: pdf(168.78 KB) Additional Information: full citation, abstract, references, citings, index terms

Numerous systems have been designed which use virtualization to subdivide the ample resources of a modern computer. Some require specialized hardware, or cannot support commodity operating systems. Some target 100% binary compatibility at the expense of performance. Others sacrifice security or functionality for speed. Few offer resource isolation or performance guarantees; most provide only best-effort provisioning, risking denial of service.This paper presents Xen, an x86 virtual machine monit ...

Keywords: hypervisors, paravirtualization, virtual machine monitors

12 MCTS nucleus: philosophy and praxis
Ralph R. Brown
January 1976 **ACM SIGOPS Operating Systems Review**, Volume 10 Issue 1
Publisher: ACM Press
Full text available: pdf(1.50 MB) Additional Information: full citation, abstract, references

The goals of the total MCTS System are stated in terms of capabilities which the customer tasks should be allowed to exercise. Treating these as premises, several derivative goals were deduced for the MCTS System, and these dictated the structure of the operating system and the nucleus. The deductions are exhibited, and the structure of the nucleus is derived. The body of this paper does not reference the hardware description of the computer on which the nucleus was implemented, and indeed the s ...

13 An open-source CVE for programming education: a case study: An open-source CVE for programming education: a case study
Andrew M. Phelps, Christopher A. Egert, Kevin J. Bierre, David M. Parks
July 2005 **ACM SIGGRAPH 2005 Courses SIGGRAPH '05**
Publisher: ACM Press
Full text available: pdf(7.32 MB) Additional Information: full citation, references

14 Maté: a tiny virtual machine for sensor networks
Philip Levis, David Culler
October 2002 **ACM SIGPLAN Notices , ACM SIGARCH Computer Architecture News , ACM SIGOPS Operating Systems Review , Proceedings of the 10th international conference on Architectural support for programming languages and operating systems ASPLOS-X**, Volume 37 , 30 , 36 Issue 10 , 5 , 5
Publisher: ACM Press
Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, references, citings

Composed of tens of thousands of tiny devices with very limited resources ("motes"), sensor networks are subject to novel systems problems and constraints. The large number of motes in a sensor network means that there will often be some failing nodes; networks must be easy to repopulate. Often there is no feasible method to recharge motes, so energy is a precious resource. Once deployed, a network must be reprogrammable although physically unreachable, and this reprogramming can be a significan ...

15 Scalability, performance, and real-time: Friendly virtual machines: leveraging a feedback-control model for application adaptation
Yuting Zhang, Azer Bestavros, Mina Guirguis, Ibrahim Matta, Richard West
June 2005 **Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments VEE '05**
Publisher: ACM Press
Full text available: pdf(317.34 KB) Additional Information: full citation, abstract, references, index terms

With the increased use of "Virtual Machines" (VMs) as vehicles that isolate applications running on the same host, it is necessary to devise techniques that enable multiple VMs to share underlying resources both fairly and efficiently. To that end, one common approach is to deploy complex resource management techniques in the hosting infrastructure. Alternately, in this paper, we advocate the use of self-adaptation in the VMs themselves based on feedback about resource usage and availability. Co ...

Keywords: feedback Control, friendy virtual machines, resource management

16 Applications: Aspect-oriented application-level scheduling for J2EE servers
Kenichi Kourai, Hideaki Hibino, Shigeru Chiba
March 2007 **Proceedings of the 6th international conference on Aspect-oriented software development AOSD '07**
Publisher: ACM Press
Full text available: pdf(347.70 KB) Additional Information: full citation, abstract, references, index terms

Achieving sufficient execution performance is a challenging goal of software development. Unfortunately, violating performance requirements is often revealed at a late stage of the development. Fixing a performance problem at such a late stage is difficult in terms of cost and time. To solve this problem, this paper presents *QoSWeaver*, which provides aspect-oriented application-level scheduling. QoSWeaver weaves scheduling code written in an aspect into application code. The scheduling co ...

Keywords: QoS, case study, performance tuning, pointcut generator

17 An implementation scheme for a virtual machine monitor to be realized on user - microprogrammable minicomputers
B. D. Shriver, J. W. Anderson, L. J. Waguespack, D. M. Hyams, R. A. Bombet
October 1976 **Proceedings of the annual conference ACM 76**
Publisher: ACM Press
Full text available: pdf(654.60 KB) Additional Information: full citation, abstract, references, citings, index terms

A virtual machine monitor allows several different operating systems to run concurrently on the same machine. This paper presents the description of a virtual machine monitor and its support structure which can be implemented on a microprogrammable minicomputer or a distributed network of such machines. In our approach, all storage, transformational, input, and output resources of the system are accessed through a mapping mechanism. The design and implementation methodology for an actual re ...

18 Compiling and program transformations: Schedulable persistence system for real-time applications in virtual machine
Okehee Goh, Yann-Hang Lee, Ziad Kaakani
October 2006 **Proceedings of the 6th ACM & IEEE International conference on Embedded software EMSOFT '06**
Publisher: ACM Press
Full text available: pdf(254.72 KB) Additional Information: full citation, abstract, references, index terms

Persistence in applications saves a computation state that can be used to facilitate system recovery upon failures. As we begin to adopt virtual execution environments (VMs) for mission-critical real-time embedded applications, persistence service will become an essential part of VM to ensure high availability of the systems.In this paper, we focus in a schedulable persistence system in VMs and show a prototype persistence system

constructed on CLI 's open source platform, MONO. By employing obj ...

Keywords: CLI, checkpoint/recovery, real-time applications, schedulable persistence system, virtual machine

19 Real-time paradigms: The reactive programming approach on top of Java/J2ME
Jean-Ferdy Susini
October 2006 **Proceedings of the 4th international workshop on Java technologies for real-time and embedded systems JTRES '06**
Publisher: ACM Press
Full text available: pdf(618.32 KB) Additional Information: full citation, abstract, references, index terms

Concurrent design facilitate the programming of interactive-applications such as games or simulations of virtual worlds. Java has popularized the use of multithreaded programming to address multiple concurrency issues in applications. However, threads are not always fine-grained enough to be successfully applied in all circumstances, especially when it comes to programming on customer electronic devices such as mobile phone or personal digital assistant. The lack of resources (memory, processing ...

Keywords: J2ME, concurrent programming, embedded systems, java, multi-clock reactive systems, reactive programming approach, synchronous-asynchronous interaction

20 The impact of operating system scheduling policies and synchronization methods of performance of parallel applications
Anoop Gupta, Andrew Tucker, Shigeru Urushibara
April 1991 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1991 ACM SIGMETRICS conference on Measurement and modeling of computer systems SIGMETRICS '91**, Volume 19 Issue 1
Publisher: ACM Press
Full text available: pdf(1.61 MB) Additional Information: full citation, abstract, references, citings, index terms

Shared-memory multiprocessors are frequently used as compute servers with multiple parallel applications executing at the same time. In such environments, the efficiency of a parallel application can be significantly affected by the operating system scheduling policy. In this paper, we use detailed simulation studies to evaluate the performance of several different scheduling strategies, These include regular priority scheduling, coscheduling or gang scheduling, process control with processor pa ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Terms used: virtual timer Found 27 of 204,472

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window
Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 27 Result page: 1 2 next

Relevance scale

1 Dynamic instrumentation of threaded applications
Zhichen Xu, Barton P. Miller, Oscar Naim
May 1999 **ACM SIGPLAN Notices , Proceedings of the seventh ACM SIGPLAN symposium on Principles and practice of parallel programming PPoPP '99,** Volume 34 Issue 8
Publisher: ACM Press
Full text available: pdf(2.85 MB) Additional Information: full citation, abstract, references, citings, index terms

The use of threads is becoming commonplace in both sequential and parallel programs. This paper describes our design and initial experience with non-trace based performance instrumentation techniques for threaded programs. Our goal is to provide detailed performance data while maintaining control of instrumentation costs. We have extended Paradyn's dynamic instrumentation (which can instrument programs without recompiling or relinking) to handle threaded programs.Controlling instrumentation cost ...

2 Communication and computation performance of the CM-5
T. T. Kwan, B. K. Totty, D. A. Reed
December 1993 **Proceedings of the 1993 ACM/IEEE conference on Supercomputing Supercomputing '93**
Publisher: ACM Press
Full text available: pdf(919.49 KB) Additional Information: full citation, references, citings, index terms

3 Mechanisms for detecting and handling timing errors
David B. Stewart, Pradeep K. Khosla
January 1997 **Communications of the ACM,** Volume 40 Issue 1
Publisher: ACM Press
Full text available: pdf(293.93 KB) Additional Information: full citation, references, citings, index terms

4 Application to communication systems: Optimizing an OBS scheduler buffer
Andrew Zalesky
October 2006 **Proceedings of the 1st international conference on Performance evaluation methodolgies and tools valuetools '06**
Publisher: ACM Press
Full text available: pdf(197.35 KB) Additional Information: full citation, abstract, references

We dimension a buffer in which headers are enqueued during times an OBS scheduler is overwhelmed. Too large a buffer requires a prolonged offset period to counterbalance longer per-hop worst-case header queueing delays, resulting in reduced throughput and increased packet loss due to edge buffer overflow. While too small a buffer cannot absorb enough variability in the header arrival process, resulting in increased burst lost due to a

greater number of headers arriving to find a full buffer. For ...

Keywords: all-optical switching, blocking probability, optical burst switching, scheduler, stochastic performance modeling

5 Experience with a performance analyzer for multithreaded applications
Gilbert J. Hansen, Charles A. Linthicum, Gary Brooks
November 1990 **Proceedings of the 1990 ACM/IEEE conference on Supercomputing Supercomputing '90**
Publisher: IEEE Computer Society
Full text available: pdf(771.61 KB) Additional Information: full citation, abstract, references

Determining the effectiveness of parallelization requires performance data about elapsed process time and total CPU time. Furthermore, it is desirable not to have to run a parallel application in a stand-alone environment in order to obtain the profile. This paper describes the CONVEX® performance analyzer, ***CXpa***™, with the capability to monitor parallel regions of code, in particular loops, executed in a time-sharing environment. The means by which profiling information i ...

6 Portability and the UNIX operating system
Daniel A. Cañas, Laura M. Esquivel
April 1988 **ACM SIGOPS Operating Systems Review,** Volume 22 Issue 2
Publisher: ACM Press
Full text available: pdf(1.13 MB) Additional Information: full citation, abstract, index terms

This paper studies the compatibility degree between two different UNIX² environments. The porting of ConcurrenC (concurrent version of the C programming language) from a VAX³ computer under UNIX 4.2BSD to an AT&T UNIX PC running UNIX System V was used as a testbed.ConcurrenC is analyzed to highlight the existence of the 4.2BSD features missing in UNIX System V. These characteristics are mainly related to system calls used for interrupt handling, and for process sync ...

7 Evaluating the memory overhead required for COMA architectures
T. Joe, J. L. Hennessy
April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture ISCA '94,** Volume 22 Issue 2
Publisher: IEEE Computer Society Press, ACM Press
Full text available: pdf(1.31 MB) Additional Information: full citation, abstract, references, citings, index terms

Cache only memory architectures (COMA) have an inherent memory overhead due to the organization of main memory as a large cache called an attraction memory. This overhead consists of memory left unallocated for performance reasons as well as additional physical memory required due to the cache organization of memory. In this work, we examine the effect of data reshuffling and data replication on the memory overhead. Data reshuffling occurs when space needs to be allocated to store a remote memor ...

8 Comparative performance evaluation of cache-coherent NUMA and COMA architectures
Per Stenström, Truman Joe, Anoop Gupta
April 1992 **ACM SIGARCH Computer Architecture News , Proceedings of the 19th annual international symposium on Computer architecture ISCA '92,** Volume 20 Issue 2
Publisher: ACM Press
Full text available: pdf(1.52 MB) Additional Information: full citation, abstract, references, citings, index terms

Two interesting variations of large-scale shared-memory machines that have recently emerged are cache-coherent non-uniform-memory-access machines (CC-NUMA) and cache-only memory architectures (COMA). They both have distributed main memory and use directory-based cache coherence. Unlike CC-NUMA, however, COMA machines

automatically migrate and replicate data at the main-memory level in cache-line sized chunks. This paper compares the performance of these two classes ...

9 Posters: Lowering radio duty cycle through temperature compensated timing
Joakim Arfvidsson, Eric Park, Philip Levis
October 2006 **Proceedings of the 4th international conference on Embedded networked sensor systems SenSys '06**
Publisher: ACM Press
Full text available: pdf(94.76 KB) Additional Information: full citation, references, index terms

Keywords: clock drift, sensor networks, temperature compensation, time synchronization

10 The impact of operating system scheduling policies and synchronization methods of performance of parallel applications
Anoop Gupta, Andrew Tucker, Shigeru Urushibara
April 1991 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1991 ACM SIGMETRICS conference on Measurement and modeling of computer systems SIGMETRICS '91,** Volume 19 Issue 1
Publisher: ACM Press
Full text available: pdf(1.91 MB) Additional Information: full citation, abstract, references, citings, index terms

Shared-memory multiprocessors are frequently used as compute servers with multiple parallel applications executing at the same time. In such environments, the efficiency of a parallel application can be significantly affected by the operating system scheduling policy. In this paper, we use detailed simulation studies to evaluate the performance of several different scheduling strategies, These include regular priority scheduling, coscheduling or gang scheduling, process control with processor pa ...

11 Virtualization and operating systems. FoxyTechnique: tricking operating system policies with a virtual machine monitor
Hiroshi Yamada, Kenji Kono
June 2007 **Proceedings of the 3rd international conference on Virtual execution environments VEE '07**
Publisher: ACM Press
Full text available: pdf(2.21 MB) Additional Information: full citation, abstract, references, index terms

Integrating new resource management policies into operating systems (OSes) is an ongoing process. Despite innovative policy proposals being developed, it is quite difficult to deploy a new one widely because it is difficult, costly and often impractical endeavor to modify existing OSes to integrate a new policy. To address this problem, we explore the possibility of using virtual machine technology to incorporate a new policy into an existing OS without the need to make any changes to it. Thi ...

Keywords: interference, resource management, virtual machine

12 Design and implementation of a framework for efficient and programmable sensor networks
Athanassios Boulis, Chih-Chieh Han, Mani B. Srivastava
May 2003 **Proceedings of the 1st international conference on Mobile systems, applications and services MobiSys '03**
Publisher: ACM Press
Full text available: pdf(303.56 KB) Additional Information: full citation, abstract, references, cited by, index terms

Wireless ad hoc sensor networks have emerged as one of the key growth areas for wireless networking and computing technologies. So far these networks/systems have been designed with static and custom architectures for specific tasks, thus providing

inflexible operation and interaction capabilities. Our vision is to create sensor networks that are open to multiple transient users with dynamic needs. Working towards this vision, we propose a framework to define and support lightweight and mobile c ...

13 Virtual machines: ReVirt: enabling intrusion analysis through virtual-machine logging and replay
George W. Dunlap, Samuel T. King, Sukru Cinar, Murtaza A. Basrai, Peter M. Chen
December 2002 **ACM SIGOPS Operating Systems Review,** Volume 36 Issue SI
Publisher: ACM Press
Full text available: pdf(1.56 MB) Additional Information: full citation, abstract, references, cited by, index terms

Current system loggers have two problems: they depend on the integrity of the operating system being logged, and they do not save sufficient information to replay and analyze attacks that include any non-deterministic events. ReVirt removes the dependency on the target operating system by moving it into a virtual machine and logging below the virtual machine. This allows ReVirt to replay the system's execution before, during, and after an intruder compromises the system, even if the intruder rep ...

14 Virtual machines: Scale and performance in the Denali isolation kernel
Andrew Whitaker, Marianne Shaw, Steven D. Gribble
December 2002 **ACM SIGOPS Operating Systems Review,** Volume 36 Issue SI
Publisher: ACM Press
Full text available: pdf(1.31 MB) Additional Information: full citation, abstract, references, cited by, index terms

This paper describes the Denali isolation kernel, an operating system architecture that safely multiplexes a large number of untrusted Internet services on shared hardware. Denali's goal is to allow new Internet services to be "pushed" into third party infrastructure, relieving Internet service authors from the burden of acquiring and maintaining physical infrastructure. Our isolation kernel exposes a virtual machine abstraction, but unlike conventional virtual machine monitors, Denali does not ...

15 PL/I program efficiency
Michael McNeil, William Tracz
June 1980 **ACM SIGPLAN Notices,** Volume 15 Issue 6
Publisher: ACM Press
Full text available: pdf(1.09 MB) Additional Information: full citation, abstract, references, citings

All PL/I Programmers should be aware of and genuinely concerned about PL/I Program efficiency. This paper addresses the following question:"How do you write a PL/I program which the PL/I Complier will reduce to the smallest and fastest executing machine language module?" The real world payoffs of knowing how the PL/I Optizing Compiler handles different syntactical representations of similar semantic relationships with respect to code generation and storage allocation can increase program runtime ...

16 Xen and the art of virtualization
Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, Andrew Warfield
October 2003 **ACM SIGOPS Operating Systems Review , Proceedings of the nineteenth ACM symposium on Operating systems principles SOSP '03,** Volume 37 Issue 5
Publisher: ACM Press
Full text available: pdf(168.78 KB) Additional Information: full citation, abstract, references, citings, index terms

Numerous systems have been designed which use virtualization to subdivide the ample resources of a modern computer. Some require specialized hardware, or cannot support commodity operating systems. Some target 100% binary compatibility at the expense of performance. Others sacrifice security or functionality for speed. Few offer resource isolation or performance guarantees; most provide only best-effort provisioning, risking denial of service.This paper presents Xen, an x86 virtual machine monit ...

Keywords: hypervisors, paravirtualization, virtual machine monitors

17 Process exchange on the PR1ME family of computers
Edward A. Feustel
March 1984 **ACM SIGARCH Computer Architecture News**, Volume 12 Issue 1
Publisher: ACM Press
Full text available: pdf(751.77 KB) Additional Information: full citation, abstract, references

A high speed mechanism for process exchange is essential in a time sharing system based on the use of many processes. The automated process exchange mechanism on the Model P400 and the 50 Series of PR1ME machines is described. Typical timing for operations using the process exchange mechanism on the P750 is given. Its use in PRIMOS is explained and validated.

18 Modeling/simulation: Modeling virtual object behavior within virtual environment
Gun A. Lee, Gerard Jounghyun Kim, Chan-Mo Park
November 2002 **Proceedings of the ACM symposium on Virtual reality software and technology VRST '02**
Publisher: ACM Press
Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, references, citings, index terms

Development of virtual reality systems requires iterations of specification, implementation and evaluation. Since correct evaluations of immersive VR systems require the tedious process of wearing many devices, there exist both temporal and spatial gaps between the implementation and evaluation stage, and this usually causes delay and inefficiency in the development process. In order to overcome this gap, there have been several approaches to constructing or modeling the physical aspects of the ...

Keywords: 3D interaction, interactive behavior modeling, programming by demonstration, virtual environment, virtual object

19 A virtual machine emulator for performance evaluation
M. D. Canon, D. H. Fritz, J. H. Howard, T. D. Howell, M. F. Mitoma, J. Rodriquez-Rosell
February 1980 **Communications of the ACM**, Volume 23 Issue 2
Publisher: ACM Press
Full text available: pdf(865.59 KB) Additional Information: full citation, references, citings

Keywords: computer system simulation, performance evaluation, virtual machines

20 The time problem
Russell M. Clapp, Trevor Mudge
January 1990 **ACM SIGAda Ada Letters , Proceedings of the working group on Ada performance issues 1990**, Volume X Issue 3
Publisher: ACM Press
Full text available: pdf(631.56 KB) Additional Information: full citation, index terms

Results 1 - 20 of 27 Result page: **1** 2 next

Terms used: **virtual time** Found 931 of 204,472

Sort results by relevance
Display results expanded form

Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown
Relevance scale

1 Virtual time
David R. Jefferson
July 1985 **ACM Transactions on Programming Languages and Systems (TOPLAS),** Volume 7 Issue 3
Publisher: ACM Press
Full text available: pdf(1.82 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Virtual time is a new paradigm for organizing and synchronizing distributed systems which can be applied to such problems as distributed discrete event simulation and distributed database concurrency control. Virtual time provides a flexible abstraction of real time in much the same way that virtual memory provides an abstraction of real memory. It is implemented using the Time Warp mechanism, a synchronization protocol distinguished by its reliance on lookahead-rollback, a ...

2 The virtual time machine
R. M. Fujimoto
March 1989 **Proceedings of the first annual ACM symposium on Parallel algorithms and architectures SPAA '89**
Publisher: ACM Press
Full text available: pdf(1.52 MB) Additional Information: full citation, citings, index terms

3 Borrowed-virtual-time (BVT) scheduling: supporting latency-sensitive threads in a general-purpose scheduler
Kenneth J. Duda, David R. Cheriton
December 1999 **ACM SIGOPS Operating Systems Review , Proceedings of the seventeenth ACM symposium on Operating systems principles SOSP '99,** Volume 33 Issue 5
Publisher: ACM Press
Full text available: pdf(1.81 MB) Additional Information: full citation, abstract, references, citings, index terms

Systems need to run a larger and more diverse set of applications, from real-time to interactive to batch, on uniprocessor and multiprocessor platforms. However, most schedulers either do not address latency requirements or are specialized to complex real-time paradigms, limiting their applicability to general-purpose systems.In this paper, we present *Borrowed-Virtual-Time (BVT) Scheduling,* showing that it provides low-latency for real-time and interactive applications yet weighted sharin ...

4 Virtual time II: storage management in conservative and optimistic systems
David Jefferson
August 1990 **Proceedings of the ninth annual ACM symposium on Principles of distributed computing PODC '90**

Publisher: ACM Press
Full text available: pdf(1.53 MB) Additional Information: full citation, references, citings, index terms

5 An algorithm for minimally latent global virtual time
Alexander I. Tomlinson, Vijay K. Garg
July 1993 **ACM SIGSIM Simulation Digest , Proceedings of the seventh workshop on Parallel and distributed simulation PADS '93,** Volume 23 Issue 1
Publisher: ACM Press
Full text available: pdf(773.05 KB) Additional Information: full citation, abstract, references, citings, index terms

Global virtual time (GVT) is used in distributed simulations to reclaim memory, commit output, detect termination, and handle errors. It is a global function that is computed many times during the course of a simulation. A small GVT latency (delay between its occurrence and detection) allows for more efficient use of resources. We present an algorithm which minimizes the latency, and we prove its correctness. The algorithm is unique in that a target virtual time (TVT) is predetermined by an ...

6 Computing global virtual time in shared-memory multiprocessors
Richard M. Fujimoto, Maria Hybinette
October 1997 **ACM Transactions on Modeling and Computer Simulation (TOMACS),** Volume 7 Issue 4
Publisher: ACM Press
Full text available: pdf(135.72 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Global virtual time (GVT) is used in the Time Warp synchronization mechanism to perform irrevocable operations such as I/O and to reclaim storage. Most existing algorithms for computing GVT assume a message-passing programming model. Here, GVT computation is examined in the context of a shared-memory model. We observe that computation of GVT is much simpler in shared-memory multiprocessors because these machines normally guarantee that no two processors will observe a set of memory operatio ...

7 Global Virtual Time and distributed synchronization
Jeffrey S. Steinman, Craig A. Lee, Linda F. Wilson, David M. Nicol
July 1995 **ACM SIGSIM Simulation Digest , Proceedings of the ninth workshop on Parallel and distributed simulation PADS '95,** Volume 25 Issue 1
Publisher: IEEE Computer Society, ACM Press
Full text available: pdf(1.46 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Global Virtual Time (GVT) is the fundamental synchronization concept in optimistic simulations. It is defined as the earliest time tag within the set of unprocessed pending events in distributed simulation. A number of techniques for determining GVT have been proposed in recent years, each having their own intrinsic properties. However, most of these techniques either focus on specific types of simulation problems or assume specific hardware support. This paper specifically addresses the GV ...

Keywords: GVT computation, SPEEDES GVT, SPEEDES framework, Synchronous Parallel Environment for Emulation and Discrete-Event Simulation framework, digital simulation, distributed simulation, distributed synchronization, efficiency, event processing, flow control, fundamental synchronization concept, global reduction operations, global virtual time, interactive support, message passing, optimistic simulations, parallel programming, portability, real time use, real-time systems, scalability, software fault tolerance, synchronisation, unprocessed pending events

8 The virtual machine
Richard M. Fujimoto
March 1991 **ACM SIGARCH Computer Architecture News,** Volume 19 Issue 1
Publisher: ACM Press

Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, index terms

A parallel computer architecture is proposed that is based on an optimistic style of execution. Specifically, the *Virtual Time Machine* (VTM) detects violations of data dependence constraints at runtime, and automatically recovers from them. In order to efficiently implement this mechanism, a sophisticated, two-dimensional memory system is proposed that is addressed using both a spatial and a *temporal* coordinate. Initially targeted for discrete event simulation applications, the Ion ...

9 Source level debugging of automatically parallelized code
Robert Cohn
December 1991 **ACM SIGPLAN Notices , Proceedings of the 1991 ACM/ONR workshop on Parallel and distributed debugging PADD '91,** Volume 26 Issue 12
Publisher: ACM Press
Full text available: pdf(1.34 MB) Additional Information: full citation, references, citings, index terms

10 Bounds and approximations for self-initiating distributed simulation without lookahead
Robert E. Felderman, Leonard Kleinrock
October 1991 **ACM Transactions on Modeling and Computer Simulation (TOMACS),** Volume 1 Issue 4
Publisher: ACM Press
Full text available: pdf(1.14 MB) Additional Information: full citation, abstract, references, citings, index terms, review

We provide upper and lower bounds and an approximation for speedup of an optimistic self-initiated distributed simulation using a very simple model. We assume an arbitrary number of processors and a uniform connection topology. By showing that the lower bound increases essentially linearly with P, the number of processors, we find that the optimistic approach scales well as P increases. The model tracks the progress of Global Virtual Time (GVT) and eliminat ...

Keywords: Global Virtual Time, Time Warp, bounds, discrete-event simulation, distributed processing, optimal, optimistic simulation, parallel processing, performance analysis, speedup, virtual time

11 Time warp operating system
D. Jefferson, B. Beckman, F. Wieland, L. Blume, M. Diloreto
November 1987 **ACM SIGOPS Operating Systems Review , Proceedings of the eleventh ACM Symposium on Operating systems principles SOSP '87,** Volume 21 Issue 5
Publisher: ACM Press
Full text available: pdf(1.52 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes the Time Warp Operating System, under development for three years at the Jet Propulsion Laboratory for the Caltech Mark III Hypercube multi-processor. Its primary goal is concurrent execution of large, irregular discrete event simulations at maximum speed. It also supports any other distributed applications that are synchronized by virtual time.The Time Warp Operating System includes a complete implementation of the Time Warp mechanism, and is a substantial d ...

12 A hierarchical fair service curve algorithm for link-sharing, real-time, and priority services
Ion Stoica, Hui Zhang, T. S. Eugene Ng
April 2000 **IEEE/ACM Transactions on Networking (TON),** Volume 8 Issue 2
Publisher: IEEE Press
Full text available: pdf(278.79 KB) Additional Information: full citation, references, citings, index terms

Keywords: fairness, link-sharing, packet scheduling, quality of service (QoS), real-time

13 A hierarchical fair service curve algorithm for link-sharing, real-time and priority services
Ion Stoica, Hui Zhang, T. S. Eugene Ng
October 1997 **ACM SIGCOMM Computer Communication Review , Proceedings of the ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communication SIGCOMM '97,** Volume 27 Issue 4
Publisher: ACM Press
Full text available: pdf(2.25 MB) Additional Information: full citation, abstract, references, citings, index terms

In this paper, we study hierarchical resource management models and algorithms that support both link-sharing and guaranteed real-time services with decoupled delay (priority) and bandwidth allocation. We extend the service curve based QoS model, which defines both delay and bandwidth requirements of a class, to include fairness, which is important for the integration of real-time and hierarchical link-sharing services. The resulting *Fair Service Curve link-sharing* model formalizes the go ...

14 Hierarchical packet fair queueing algorithms
Jon C. R. Bennett, Hui Zhang
October 1997 **IEEE/ACM Transactions on Networking (TON),** Volume 5 Issue 5
Publisher: IEEE Press
Full text available: pdf(326.41 KB) Additional Information: full citation, references, citings, index terms, review

Keywords: hierarchical packet scheduling, link-sharing, quality of service, real-time, resource management

15 Decoupling QoS control from core routers: a novel bandwidth broker architecture for scalable support of guaranteed services
Zhi-Li Zhang, Zhenhai Duan, Lixin Gao, Yiwei Thomas Hou
August 2000 **ACM SIGCOMM Computer Communication Review , Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication SIGCOMM '00,** Volume 30 Issue 4
Publisher: ACM Press
Full text available: pdf(504.39 KB) Additional Information: full citation, abstract, references, citings, index terms

For scalable support of guaranteed services that decouples the QoS control plane from the packet forwarding plane. More specifically, under this architecture, core routers do not maintain any QoS reservation states, whether per-flow or aggregate. Instead, QoS reservation states are stored at and managed by bandwidth broker(s). There are several advantages of such a bandwidth broker architecture. Among others, it relieves core routers of QoS control functions such as admissi ...

16 How to integrate shared variables in distributed simulation
Horst Mehl, Stefan Hammes
September 1995 **ACM SIGSIM Simulation Digest,** Volume 25 Issue 2
Publisher: ACM Press
Full text available: pdf(1.77 MB) Additional Information: full citation, abstract, references, index terms

Although users may want to employ shared variables when they program distributed simulation applications, almost none of the currently existing distributed simulation systems does offer this facility. In this paper, we systematically present new algorithms which provide consistent shared variables for distributed simulation applications. Basically, our approach combines known techniques to realize distributed shared memory with simulation algorithms. As there are essentially two classes of distr ...

Keywords: conservative simulation, discrete event simulation, distributed algorithms, distributed shared memory, distributed simulation, optimistic simulation, shared variables

17 Exact GPS simulation with logarithmic complexity, and its application to an optimally fair scheduler
Paolo Valente
August 2004 **ACM SIGCOMM Computer Communication Review , Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '04**, Volume 34 Issue 4
Publisher: ACM Press
Full text available: pdf(405.35 KB) Additional Information: full citation, abstract, references, index terms

Generalized Processor Sharing (GPS) is a fluid scheduling policy providing perfect fairness. The minimum deviation (lead/lag) with respect to the GPS service achievable by a packet scheduler is one packet size. To the best of our knowledge, the only packet scheduler guaranteeing such minimum deviation is Worst-case Fair Weighted Fair Queueing (WF^2Q), that requires on-line GPS simulation. Existing algorithms to perform GPS simulation have $O(N)$ complexity per packet transm ...

Keywords: computational complexity, data structures, packet scheduling, quality of service

18 Optimistic distributed simulation based on transitive dependency tracking
Om P. Damani, Yi-Min Wang, Vijay K. Garg
June 1997 **ACM SIGSIM Simulation Digest , Proceedings of the eleventh workshop on Parallel and distributed simulation PADS '97**, Volume 27 Issue 1
Publisher: IEEE Computer Society, ACM Press
Full text available: pdf(1.02 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

In traditional optimistic distributed simulation protocols, a logical process (LP) receiving a straggler rolls back and sends out anti-messages. The receiver of an anti-message may also roll back and send out more anti-messages. So a single straggler may result in a large number of anti-messages and multiple rollbacks of some LPs. In the authors' protocol, an LP receiving a straggler broadcasts its rollback. On receiving this announcement, other LPs may roll back but they do not announce their r ...

Keywords: anti-messages, dependency information, distributed recovery, logical process, memory management, message tagging, optimistic distributed simulation, optimistic distributed simulation protocols, process rollback, rollback broadcasting, straggler, time warp simulation, transitive dependency information, transitive dependency tracking

19 A virtual machine emulator for performance evaluation
M. D. Canon, D. H. Fritz, J. H. Howard, T. D. Howell, M. F. Mitoma, J. Rodriquez-Rosell
February 1980 **Communications of the ACM**, Volume 23 Issue 2
Publisher: ACM Press
Full text available: pdf(865.58 KB) Additional Information: full citation, references, citings

Keywords: computer system simulation, performance evaluation, virtual machines

20 Performance analysis of Time Warp with homogeneous processors and exponential task times
Anurag Gupta, Ian Akyildiz, Richard M. Fujimoto
April 1991 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1991 ACM SIGMETRICS conference on Measurement and modeling of**

computer systems SIGMETRICS '91, Volume 19 Issue 1
Publisher: ACM Press
Full text available: pdf(1.17 MB) Additional Information: full citation, abstract, references, citings, index terms

The behavior of n interacting processors synchronized by the "Time Warp" protocol is analyzed using a discrete state continuous time Markov chain model. The performance and dynamics of the processes are analyzed under the following assumptions: exponential task times and times-tamp increments on messages, each event message generates one new message that is sent to a randomly selected process, negligible rollback, state saving, and communication delay, unbounded message buffers, and homogeneous ...

Results 1 - 20 of 200 Result page: 1 2 3 4 5 6 7 8 9 10 next